

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	173	(shared adj lock) and (exclusive adj lock)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/17 14:51
S2	123	((shared adj lock) and (exclusive adj lock)) and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 13:16
S3	27	("4606002"   "4677550"   "4791561"   "4805099"   "4947320"   "4967341"   "5089952"   "5115392"   "5123104"   "5163148"   "5204958"   "5265244"   "5386557"   "5430869"   "5493728"   "5504888"   "5555389"   "5630124"   "5721909"   "5758356"   "5794241"   "5835757"   "5842196"   "5884327"   "5983225"   "6052695"   "6112209").PN.	USPAT	OR	ON	2004/06/17 15:08
S4	191	(local adj lock) and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/17 15:16
S5	33	((local adj lock) and (@ad<="20001220" @ad<="20001220")) and (exclusive adj lock)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 10:19
S6	5	"5805900".URPN.	USPAT	OR	ON	2004/06/17 15:56
S7	17	("4965719"   "5175852"   "5202971"   "5226159"   "5327556"   "5339427"   "5410697"   "5423044"   "5442763"   "5454108"   "5537574"   "5551046"   "5596754"   "5613139"   "5634122"   "5682537"   "5734909").PN.	USPAT	OR	ON	2004/06/17 15:58
S8	27	(local adj lock adj management) and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 11:30
S9	655	707/8.ccls. and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 10:28
S10	793	707/201.ccls. and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 10:28

S11	53	((707/8.ccls. and (@ad<="20001220" @ad<="20001220")) or (707/201. ccls. and (@ad<="20001220" @ad<="20001220"))) and (resource adj manager)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 10:28
S12	69	(local adj lock adj manager) and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 13:09
S13	86	(distribute\$ adj lock adj manager) and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 13:09
S14	228	((read adj lock) and (write adj lock)) and (@ad<="20001220" @ad<="20001220")	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 13:16
S15	43	((read adj lock) and (write adj lock)) and (@ad<="20001220" @ad<="20001220")) and ((local private) adj lock)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2004/06/18 13:17
S16	121	autoplay	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/02/10 14:55
S17	1	"6466952"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2005/02/10 14:55

Searching for **PHRASE local lock**.

Restrict to: [Header](#) [Title](#) Order by: [Expected citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

23 documents found. **Order: number of citations.**

[Adaptive Software Cache Management for Distributed.. - Bennett, Carter.. \(1990\) \(Correct\) \(69 citations\)](#)  
a local proxy for the distributed lock, and the **local lock** server arbitrates with the remote lock servers  
[www-brazos.rice.edu/jkb/papers/ieee95.ps](http://www-brazos.rice.edu/jkb/papers/ieee95.ps)

[Decoding Choice Encodings - Nestmann, Pierce \(1999\) \(Correct\) \(29 citations\)](#)  
runs a mutual exclusion protocol, installing a **local lock**-a message that carries a special name-on  
Q j#J Branch I R j jinstalls a **local lock**-a message on I that carries a boolean  
[www.brics.dk/RS/99/Ref/BRICS-RS-99-Ref/.../RS/99/42/BRICS-RS-99-42.ps.gz](http://www.brics.dk/RS/99/Ref/BRICS-RS-99-Ref/.../RS/99/42/BRICS-RS-99-42.ps.gz)

[Client-Server Caching Revisited - Michael Franklin \(1992\) \(Correct\) \(12 citations\)](#)  
lock caching is optimistic in the sense that a **local lock** at one site does not guarantee the absence of  
does not guarantee the absence of conflicting **local locks** at other sites. The three O2PL-based  
[www.cs.umd.edu/users/franklin/papers/iwdom.ps.gz](http://www.cs.umd.edu/users/franklin/papers/iwdom.ps.gz)

[Priority Inheritance Spin Locks for Multiprocessor Real-Time.. - Cai-Dong Wang \(1996\) \(Correct\) \(5 citations\)](#)  
global shared variables. shared var L1, L2: **Lock local** variables (allocated for each processor)  
[www.ertl.ics.tut.ac.jp/~hiro/papers/ispan96.ps.Z](http://www.ertl.ics.tut.ac.jp/~hiro/papers/ispan96.ps.Z)

[Improving the Efficiency of Object Invocations by Dynamic Object .. - Windisch \(1995\) \(Correct\) \(3 citations\)](#)  
reader-writer synchronization. To allow for **local lock** acquisition each time a node acquires a lock it  
[www13.informatik.tu-muenchen.de/forschung/papers/PDPTA95.ps](http://www13.informatik.tu-muenchen.de/forschung/papers/PDPTA95.ps)

[Orphan Detection in the Argus System - Walker \(1984\) \(Correct\) \(2 citations\)](#)  
Subaction Commit 4.2.9 Prepare Messages 4.2.10 **Local Lock** Propagation 9 12 12 14 16 19 19 21 24  
the Growth of Done 4.4.3 D-list-map 4.4.4 **Local Lock** Propagation 4.5 Orphan Extermination 4.5.1 How  
[www.lcs.mit.edu/publications/pubs/ps/tr-326.ps](http://www.lcs.mit.edu/publications/pubs/ps/tr-326.ps)

[Fine-granularity Locking and Client-Based Logging.. - Panagos, Biliris.. \(1996\) \(Correct\) \(2 citations\)](#)  
two-phase locking protocol. Each client has a **local lock** manager #LLM# that caches all acquired locks  
[www.research.att.com/~biliris/pubs/papers/96\\_edbt.pdf](http://www.research.att.com/~biliris/pubs/papers/96_edbt.pdf)

[Scheduling Transactions on Distributed Systems with the VPL.. - Kühn, Liu, Pohlai \(1994\) \(Correct\) \(1 citation\)](#)  
compensation. A global transaction must not **lock local** (database) systems over a long period of time.  
[www.complang.tuwien.ac.at/eva/esda-94-camera-ready.ps.Z](http://www.complang.tuwien.ac.at/eva/esda-94-camera-ready.ps.Z)

[Fair Multi-Branch Locking of Several Locks - Fleiner, Philippsen \(1997\) \(Correct\) \(1 citation\)](#)  
and starvation. Both left frk and right frk are **local lock** objects that are specific for each philosopher  
[i43fs2.ira.uka.de/~philipp/locks.ps.gz](http://i43fs2.ira.uka.de/~philipp/locks.ps.gz)

[A Centralized Token-Based Algorithm for Distributed Mutual.. - Edward Felten \(1992\) \(Correct\) \(1 citation\)](#)  
In these algorithms, each node maintains its own **local lock** that can be given to one process at a time. In  
attain mutual exclusion, a process must obtain **local locks** on a set of nodes called a quorum [6, 7]  
[www.research.att.com/~misha/otherPubs/centrToken.ps.gz](http://www.research.att.com/~misha/otherPubs/centrToken.ps.gz)

[The TACO Protocol Processor Simulation Environment - Virtanen, Lilius \(2001\) \(Correct\)](#)  
the interconnection network. GL=global lock, LL=**local lock** request, SQ=squash. data buses, controlled by  
[www.tucs.fi/Pub/2001/Proc/pViLib.pdf](http://www.tucs.fi/Pub/2001/Proc/pViLib.pdf)

[Research Memoranda - Fashions Habits And \(Correct\)](#)  
David, 1985)However, it does not explain '**local lock** ins' such as different clothing styles in  
the results for a typical imitators run. **Local lock**-ins can be observed, where a few products ends  
[zappa.ubvu.vu.nl/20000031.pdf](http://zappa.ubvu.vu.nl/20000031.pdf)

Clustered/Distributed File System - Calypso File System (Correct)

Acquire locks Request lock to token client (**local lock** manager LLM) To grant a lock, an LLM should  
[sslab.postech.ac.kr/research/cfs\\_calypso1.ps](http://sslab.postech.ac.kr/research/cfs_calypso1.ps)

Atomicity, Serialization and Recovery in the Island-Based.. - Minwen Ji Princeton (Correct)

in each island, which is guaranteed with a **local lock** in that island. 2) A coordinator does not  
[www.cs.princeton.edu/~mji/etabstract.pdf](http://www.cs.princeton.edu/~mji/etabstract.pdf)

A Processor Architecture for the TACO Protocol.. - Virtanen, Lilius.. (2000) (Correct)

interconnection network. GL =global lock, LL =**local lock** request, SQ =squash. units that are able to  
[www.tucs.fi/Pub/2000/Proc/pViLiWea.pdf](http://www.tucs.fi/Pub/2000/Proc/pViLiWea.pdf)

Logging and Crash Recovery in Shared-Disk Parallel Database.. - Bozas, Kober (1998) (Correct)

protocol for efficient transaction processing. A **local lock** manager on each node administers **local lock**  
A **local lock** manager on each node administers **local lock** requests. Committed modifications to a data  
[ftp.leo.org/pub/comp/doc/techreports/tum/informatik/report/1998/TUM-I9812.ps.gz](http://ftp.leo.org/pub/comp/doc/techreports/tum/informatik/report/1998/TUM-I9812.ps.gz)

*First 20 documents* [Next 20](#)

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer - Copyright [NEC](#) and [IST](#)

## Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

## Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

## Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

## Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

Your search matched **1** of **1045422** documents.  
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

### Refine This Search:

You may refine your search by editing the current search expression or entering new one in the text box.

local lock

**Search**

☐ Check to search within this result set

### Results Key:

**JNL** = Journal or Magazine **CNF** = Conference **STD** = Standard

### 1 Private locking and distributed cache management

*Lomet, D.;*

Parallel and Distributed Information Systems, 1994., Proceedings of the Third International Conference on , 28-30 Sept. 1994

Pages:151 - 159

[\[Abstract\]](#)

[\[PDF Full-Text \(844 KB\)\]](#)

**IEEE CNF**

## Print Format

## Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

## Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

## Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

## Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

Your search matched **8** of **1045422** documents.  
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

### Refine This Search:

You may refine your search by editing the current search expression or entering new one in the text box.



☐ Check to search within this result set

### Results Key:

**JNL** = Journal or Magazine   **CNF** = Conference   **STD** = Standard

#### 1 Efficient implementation of mutual exclusion locks in large multiprocessors

*Nian-Feng Tzeng; Fu, S.S.;*

Parallel Processing Symposium, 1995. Proceedings., 9th International , 25-28 Ap 1995

Pages:270 - 275

[\[Abstract\]](#)   [\[PDF Full-Text \(560 KB\)\]](#)   IEEE CNF

#### 2 Performance analysis of static locking in replicated distributed databases systems

*Kuang, Y.; Mukkamala, R.;*

Southeastcon '91., IEEE Proceedings of , 7-10 April 1991

Pages:698 - 701 vol.2

[\[Abstract\]](#)   [\[PDF Full-Text \(216 KB\)\]](#)   IEEE CNF

#### 3 Using atomic data structures for parallel simulation

*Barth, P.S.;*

Scalable High Performance Computing Conference, 1992. SHPCC-92. Proceedings , 26-29 April 1992

Pages:30 - 37

[\[Abstract\]](#)   [\[PDF Full-Text \(432 KB\)\]](#)   IEEE CNF

#### 4 A circular list based mutual exclusion scheme for large shared-memory multiprocessors

*Fu, S.S.; Nian-Feng Tzeng;*

Parallel and Distributed Systems, IEEE Transactions on , Volume: 8 , Issue: 6 , June 1997

Pages:628 - 639

[\[Abstract\]](#) [\[PDF Full-Text \(324 KB\)\]](#) **IEEE JNL**

---

**5 Optional instant locking in distributed collaborative graphics editing systems**

*Chen, D.; Chengzheng Sun;*

Parallel and Distributed Systems, 2001. ICPADS 2001. Proceedings. Eighth International Conference on , 26-29 June 2001

Pages:109 - 116

[\[Abstract\]](#) [\[PDF Full-Text \(780 KB\)\]](#) **IEEE CNF**

---

**6 Evaluating the impact of the programming model on the performance and complexity of software DSM systems**

*Seidel, C.B.; Bianchini, R.; Amorim, C.L.;*

Parallel Processing, 1999. Proceedings. 1999 International Conference on , 21-24 Sept. 1999

Pages:228 - 236

[\[Abstract\]](#) [\[PDF Full-Text \(104 KB\)\]](#) **IEEE CNF**

---

**7 A performance comparison of locking methods with limited wait depth**

*Thomasian, A.;*

Knowledge and Data Engineering, IEEE Transactions on , Volume: 9 , Issue: 3 , May-June 1997

Pages:421 - 434

[\[Abstract\]](#) [\[PDF Full-Text \(220 KB\)\]](#) **IEEE JNL**

---

**8 File partitioning as a means to reduce lock contention in the multiprocessor environment**

*Hoevel, L.; Panfilov, O.;*

System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on , Volume: i , 7-10 Jan. 1992

Pages:211 - 217 vol.1

[\[Abstract\]](#) [\[PDF Full-Text \(416 KB\)\]](#) **IEEE CNF**

---



[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

"local lock management"

Search

[Advanced Search](#)  
[Preferences](#)

Web

Results 1 - 7 of about 11 for "**local lock management**". (0.22 seconds)

[PDF] [Private Lock Management](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

Page 1. Private Lock Management David Lomet Digital Equipment Corporation

Cambridge Research Lab CRL November Page 2. Digital Equipment ...

[www.hpl.hp.com/techreports/Compaq-DEC/CRL-92-9.pdf](http://www.hpl.hp.com/techreports/Compaq-DEC/CRL-92-9.pdf) - [Similar pages](#)

[PS] [Private Lock Management](#)

File Format: Adobe PostScript - [View as Text](#)

... 24 REFERENCES managed locally and privately. So **local lock management** can both reduce lock overhead and improve concurrency for principals of the local LM. ...

[gatekeeper.dec.com/pub/DEC/CRL/research-reports/92.9.ps](http://gatekeeper.dec.com/pub/DEC/CRL/research-reports/92.9.ps) - [Similar pages](#)

[TITLE: \[OpenVMS\] Cookbook of Performance Slowdown, VAX and Alpha ...](#)

... spends in system functions such as system services, handling page faults, file processing, DECnet processing, handling **local lock management** and setting up ...

[starlet.deltatel.ru/~laishev/perf-cookbook.txt](http://starlet.deltatel.ru/~laishev/perf-cookbook.txt) - 71k - [Cached](#) - [Similar pages](#)

[PS] [Fine-Grained Sharing in a Page Server OODBMS](#)

File Format: Adobe PostScript - [View as Text](#)

... Extensions to multiple servers with partitioned data are straightforward, and **local lock management** can be used to allow multiple transactions on a client to ...

[www.cs.rmit.edu.au/~zahirt/Teaching/oodb/Papers/pages.ps](http://www.cs.rmit.edu.au/~zahirt/Teaching/oodb/Papers/pages.ps) - Supplemental Result - [Similar pages](#)

[IMS Family - IMS Glossary - IBM Software](#)

... **local lock management** Lock management that controls access to database records used by application programs in the same online IMS system. ...

[www.ibm.com/software/data/ims/shelf/glossary.htm](http://www.ibm.com/software/data/ims/shelf/glossary.htm) - 101k - [Cached](#) - [Similar pages](#)

[Method and system for sidefile status polling in a <a href ...](#)

... 1. Details of standard MVS or other operating system services, such as **local lock management**, subsystem invocation by interrupt or monitor, and the posting and ...

[gauss.bacon.su.se/EP/5/6/EP566964.html](http://gauss.bacon.su.se/EP/5/6/EP566964.html) - 71k - [Cached](#) - [Similar pages](#)

[Data](http://gauss.bacon.su.se/EP/5/6/EP566964.html) back-up in [Library: System Macros and Facilities](#), Volume 1. Details of standard MVS or

other operating system services such as **local lock management**, sub-system ...

[gauss.bacon.su.se/EP/6/1/EP617362.html](http://gauss.bacon.su.se/EP/6/1/EP617362.html) - 66k - [Cached](#) - [Similar pages](#)

*In order to show you the most relevant results, we have omitted some entries very similar to the 7 already displayed.*

*If you like, you can repeat the search with the omitted results included.*

"local lock management"

Search



[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2004 Google



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

**SEARCH**

THE ACM DIGITAL LIBRARY



[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used **exclusive lock** and **share lock**

Found **109** of **138,517**

Sort results  
by



[Save results to a Binder](#)

Try an [Advanced Search](#)

Display  
results



[Search Tips](#)

Try this search in [The ACM Guide](#)

☐ Open results in a new  
window

Results 1 - 20 of 109

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [next](#)

Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [Locking Protocols: From Exclusive to Shared Locks](#)

Zvi M. Kedem, Abraham Silberschatz

October 1983 **Journal of the ACM (JACM)**, Volume 30 Issue 4

Full text available: [pdf\(802.95 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

### 2 [Concurrency control: methods, performance, and analysis](#)

Alexander Thomasian

March 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 1

Full text available: [pdf\(427.18 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** Markov chains, adaptive methods, concurrency control, data contention, deadlocks, flow diagrams, load control, optimistic concurrency control, queueing network models, restart-oriented locking methods, serializability, thrashing, two-phase locking, two-phase processing, wait depth limited methods

### 3 [Transaction management I: Deadlock removal using partial rollback in database systems](#)

Donald Fussell, Zvi M. Kedem, Abraham Silberschatz

April 1981 **Proceedings of the 1981 ACM SIGMOD international conference on Management of data**


Full text available: [pdf\(1.08 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The problem of removing deadlocks from concurrent database systems using the two-phase locking protocol is considered. In particular, for systems which use no a priori information about transaction behavior in order to avoid deadlocks, it has generally been assumed necessary to totally remove and restart some transaction involved in a deadlock in order to relieve the situation. In this paper, a new approach to deadlock removal in such systems based on partial rollbacks is introduced. This approach ...

### 4 [Optimal locking integrated with operational transformation in distributed real-time group editors](#)

Chengzheng Sun, Rok Sosič

May 1999 **Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing**

Full text available:  [pdf\(1.23 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

**Keywords:** consistency maintenance, distributed computing, group editors, groupware, locking, operational transformation

5 Session 2: Concurrency control in graph protocols by using edge locks

Gael N. Buckley, Avi Silberschatz

April 1984 **Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems**

Full text available:  [pdf\(633.75 KB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#)

A large number of locking protocols use precedence relations among data items to ensure the serializability of the database system. These protocols have extended the semantics of the exclusive lock from prohibiting access to a data item to prohibiting access to an entire subgraph. In this paper we argue that combining the use of exclusive locks for these different purposes is ill conceived. We present a general theory on how these two distinct functions can be separated into the traditional lock ...

6 UIO: a uniform I/O system interface for distributed systems

David R. Cheriton

January 1987 **ACM Transactions on Computer Systems (TOCS)**, Volume 5 Issue 1

Full text available:  [pdf\(3.20 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A uniform I/O interface allows programs to be written relatively independently of specific I/O services and yet work with a wide variety of the I/O services available in a distributed environment. Ideally, the interface provides this uniform access without excessive complexity in the interface or loss of performance. However, a uniform interface does not arise from careful design of individual system interfaces alone; it requires explicit definition. In this paper, the UIO (unifo ...

7 Locking performance in centralized databases

Y. C. Tay, Nathan Goodman, R. Suri

December 1985 **ACM Transactions on Database Systems (TODS)**, Volume 10 Issue 4

Full text available:  [pdf\(3.25 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

An analytic model is used to study the performance of dynamic locking. The analysis uses only the steady-state average values of the variables. The solution to the model is given by a cubic, which has exactly one valid root for the range of parametric values that is of interest. The model's predictions agree well with simulation results for transactions that require up to twenty locks. The model separates data contention from resource contention, thus facilitating an analysis of their separ ...

8 A fast file system for UNIX

Marshall K. McKusick, William N. Joy, Samuel J. Leffler, Robert S. Fabry

August 1984 **ACM Transactions on Computer Systems (TOCS)**, Volume 2 Issue 3

Full text available:  [pdf\(1.31 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

**Keywords:** UNIX, application program interface, file system design, file system

organization, file system performance

9 The five color concurrency control protocol: non-two-phase locking in general databases

Partha Dasgupta, Zvi M. Kedem

June 1990 **ACM Transactions on Database Systems (TODS)**, Volume 15 Issue 2

Full text available:  pdf(2.20 MB)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Concurrency control protocols based on two-phase locking are a popular family of locking protocols that preserve serializability in general (unstructured) database systems. A concurrency control algorithm (for databases with no inherent structure) is presented that is practical, non two-phase, and allows varieties of serializable logs not possible with any commonly known locking schemes. All transactions are required to predeclare the data they intend to read or write. Using this informatio ...

10 Session 6: A simple analytic model for performance of exclusive locking in database systems

Nathan Goodman, Rajan Suri, Y. C. Tay

March 1983 **Proceedings of the 2nd ACM SIGACT-SIGMOD symposium on Principles of database systems**

Full text available:  pdf(909.41 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Many different algorithms have been proposed for database concurrency control, and many more can be synthesized by combining locking and timestamping. The correctness of these algorithms is already well understood, their performance is not. We need a model to help us understand, compare and control the behavior of locking and timestamping we present here a model which we hope will eventually play such a role, but which we believe is simple to understand and use.

11 Scheduling real-time transactions: a performance evaluation

Robert K. Abbott, Hector Garcia-Molina

September 1992 **ACM Transactions on Database Systems (TODS)**, Volume 17 Issue 3

Full text available:  pdf(2.93 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

**Keywords:** deadlines, locking protocols, real-time systems

12 Single sided MPI implementations for SUN MPIr

S. Booth, E. Mourao

November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available:  pdf(129.35 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

 [Publisher Site](#)

This paper describes an implementation of generic MPI-2 single-sided communications for SUN-MPI. Our implementation is layered on top of point-to-point MPI communications and therefore can be adapted to other MPI implementations. The code is designed to co-exist with other MPI-2 single-sided implementations (for example, direct use of shared memory) providing a generic fall-back implementation for those communication paths where an optimized single-sided implementation i ...

## Synchronization with multiprocessor caches

Joonwon Lee, Umakishore Ramachandran

May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture**, Volume 18 Issue 3

Full text available:  [pdf\(1.18 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Introducing private caches in bus-based shared memory multiprocessors leads to the cache consistency problem since there may be multiple copies of shared data. However, the ability to snoop on the bus coupled with the fast broadcast capability allows the design of special hardware support for synchronization. We present a new lock-based cache scheme which incorporates synchronization into the cache coherency mechanism. With this scheme high-level synchronization primitives as well as low-le ...

## 14 A mean value performance model for locking in databases: the no-waiting case

Y. C. Tay, Rajan Suri, Nathan Goodman

July 1985 **Journal of the ACM (JACM)**, Volume 32 Issue 3

Full text available:  [pdf\(2.18 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A new performance model for dynamic locking is proposed. It is based on a flow diagram and uses only the steady state average values of the variables. It is general enough to handle nonuniform access, shared locks, static locking, multiple transaction classes, and transactions of indeterminate length. The analysis is restricted to the case in which all conflicts are resolved by restarts. It has been shown elsewhere that, under certain conditions, this pure restart policy is as good as, if n ...

## 15 Load control for locking: the "half-and-half" approach

Michael J. Carey, Sanjay Krishnamurthi, Miron Livny

April 1990 **Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Full text available:  [pdf\(1.50 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A number of concurrency control performance studies have shown that, under high levels of data contention, concurrency control algorithms can exhibit thrashing behavior which is detrimental to overall system performance. In this paper, we present an approach to eliminating thrashing in the case of two-phase locking, a widely used concurrency control algorithm. Our solution, which we call the 'Half-and-Half' Algorithm, involves monitoring the state of the DBMS in order to dynamically control ...

## 16 Applying formal methods to semantic-based decomposition of transactions

Paul Ammann, Sushil Jajodia, Indrakshi Ray

June 1997 **ACM Transactions on Database Systems (TODS)**, Volume 22 Issue 2

Full text available:  [pdf\(569.45 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

In some database applications the traditional approach of serializability, in which transactions appear to execute atomically and in isolation on a consistent database state, fails to satisfy performance requirements. Although many researchers have investigated the process of decomposing transactions into steps to increase concurrency, such research typically focuses on providing algorithms necessary to implement a decomposition supplied by the database application developer and pays relat ...

**Keywords:** concurrency control, database management systems, transaction processing

17 Optional and responsive locking in collaborative graphics editing systems

David Chen, Chengzheng Sun

December 1999 **ACM SIGGROUP Bulletin**, Volume 20 Issue 3

Full text available:  pdf(430.22 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Object-based collaborative graphics editing systems allow multiple users to edit the same graphics document at the same time from multiple sites. This paper examines the use of locking to prevent the generation of conflicting operations in this type of systems. Two types of locks are examined: object and region. A locking scheme which preserves the intentions of all operations is proposed. Furthermore, the problems of lock ownership caused by concurrent operations are resolved.

18 REQUEST II — a distributed database system for local area networks

Marek Rusinkiewicz, Dimitrios Georgakopoulos

November 1999 **Proceedings of 1986 fall joint computer conference on Fall joint computer conference**

Full text available:  pdf(1.02 MB) Additional Information: [full citation](#), [references](#), [index terms](#)

19 The performance of current B-tree algorithms

Theodore Johnson, Dennis Sasha

March 1993 **ACM Transactions on Database Systems (TODS)**, Volume 18 Issue 1


Full text available:  pdf(2.87 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

**Keywords:** B-trees, concurrent B-trees, concurrent data structures, performance of concurrent algorithms

20 Replication and consistency: being lazy helps sometimes

Yuri Breitbart, Henry F. Korth

May 1997 **Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Full text available:  pdf(2.11 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 109

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Terms used **local lock**

Found 76 of 138,517

Sort results  
by

relevance




 [Save results to a Binder](#)

[Try an Advanced Search](#)

Display  
results

expanded form



 [Search Tips](#)

[Try this search in The ACM Guide](#)

☐ Open results in a new  
window

Results 1 - 20 of 76


Result page: [1](#) [2](#) [3](#) [4](#) [next](#)

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Transaction management issues in a failure-prone multidatabase system environment](#)

Yuri Breitbart, Avi Silberschatz, Glenn R. Thompson

July 1992 **The VLDB Journal — The International Journal on Very Large Data Bases**,  
Volume 1 Issue 1

Full text available:  [pdf\(2.12 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#)

This paper is concerned with the problem of integrating a number of existing, off-the-shelf local database systems into a multidatabase system that maintains consistency in the face of concurrency and failures. The major difficulties in designing such systems stem from the requirements that local transactions be allowed to execute outside the multidatabase system control, and that the various local database systems cannot participate in the execution of a global commit protocol. A scheme based o ...

**Keywords:** algorithms, deadlock recovery, performance, reliability, serializability, transaction log

2 [Reliable transaction management in a multidatabase system](#)

Yuri Breitbart, Avi Silberschatz, Glenn R. Thompson

May 1990 **ACM SIGMOD Record , Proceedings of the 1990 ACM SIGMOD international conference on Management of data**, Volume 19 Issue 2

Full text available:  [pdf\(1.33 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A model of a multidatabase system is defined in which each local DBMS uses the two-phase locking protocol. Locks are released by a global transaction only after the transaction commits or aborts at each local site. Failures may occur during the processing of transactions. We design a fault tolerant transaction management algorithm and recovery procedures that retain global database consistency. We also show that our algorithms ensure freedom from global deadlocks of any kind.

3 [Using semantic knowledge for transaction processing in a distributed database](#)

Hector Garcia-Molina

June 1983 **ACM Transactions on Database Systems (TODS)**, Volume 8 Issue 2

Full text available:  [pdf\(2.36 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper investigates how the semantic knowledge of an application can be used in a distributed database to process transactions efficiently and to avoid some of the delays


associated with failures. The main idea is to allow nonserializable schedules which preserve consistency and which are acceptable to the system users. To produce such schedules, the transaction processing mechanism receives semantic information from the users in the form of transaction semantic types, a division of tra ...

**Keywords:** concurrency control, consistency, locking, schedule, semantic knowledge, serializability

#### 4 Read-only transactions in a distributed database

Hector Garcia-Molina, Gio Wiederhold

June 1982 **ACM Transactions on Database Systems (TODS)**, Volume 7 Issue 2

Full text available:  pdf(2.16 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A read-only transaction or query is a transaction which does not modify any data. Read-only transactions could be processed with general transaction processing algorithms, but in many cases it is more efficient to process read-only transactions with special algorithms which take advantage of the knowledge that the transaction only reads. This paper defines the various consistency and currency requirements that read-only transactions may have. The processing of the different classes of read- ...

**Keywords:** R insularity, concurrency control, consistency, currency, query, read-only transaction, schedule, serializability, transaction, transaction processing algorithm

#### 5 A reliable distributed control algorithm for updating replicated databases

George Gardarin, Wesley W. Chu

November 1979 **Proceedings of the sixth symposium on Data communications**

Full text available:  pdf(779.63 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a robust, deadlock-free and distributed control algorithm for consistently updating replicated databases. This algorithm is based on local locking and time stamps on lock tables which permit detection of conflicts among transactions executed at replicated databases. Messages are exchanged in the network whenever a transaction commitment occurs, that is, at the end of every consistent step of local processing. Conflicts among remote transactions are resolved by a roll bac ...

#### 6 Affinity-based management of main memory database clusters

Minwen Ji

November 2002 **ACM Transactions on Internet Technology (TOIT)**, Volume 2 Issue 4

Full text available:  pdf(553.96 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We study management strategies for main memory database clusters that are interposed between Internet applications and back-end databases as content caches. The task of management is to allocate data across individual cache databases and to route queries to the appropriate databases for execution. The goal is to maximize effective cache capacity and to minimize synchronization cost. We propose an affinity-based management system for main memory database cLusters (*ALBUM*). *ALBUM* executes ea ...

**Keywords:** Main memory database, clustering, database administration, database cluster, file organization, query affinity, scalability

7

#### A Survey of Techniques for Synchronization and Recovery in Decentralized Computer



## Systems

Walter H. Kohler


June 1981 **ACM Computing Surveys (CSUR)**, Volume 13 Issue 2

Full text available:  [pdf\(3.33 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

### 8 Nomadic pict: correct communication infrastructure for mobile computation

Asis Unyapoth, Peter Sewell

January 2001 **ACM SIGPLAN Notices , Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 36 Issue 3

Full text available:  [pdf\(326.01 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper addresses the design and verification of infrastructure for mobile computation. In particular, we study language primitives for communication between mobile agents. They can be classified into two groups. At a low level there are *location dependent* primitives that require a programmer to know the current site of a mobile agent in order to communicate with it. At a high level there are *location independent* primitives that allow communication with a mobile agent irrespective ...

### 9 A locking protocol for resource coordination in distributed databases

Daniel A. Menasce, Gerald J. Popek, Richard R. Muntz

June 1980 **ACM Transactions on Database Systems (TODS)**, Volume 5 Issue 2

Full text available:  [pdf\(2.69 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A locking protocol to coordinate access to a distributed database and to maintain system consistency throughout normal and abnormal conditions is presented. The proposed protocol is robust in the face of crashes of any participating site, as well as communication failures. Recovery from any number of failures during normal operation or any of the recovery stages is supported. Recovery is done in such a way that maximum forward progress is achieved by the recovery procedures. Integration of ...

**Keywords:** concurrency, consistency, crash recovery, distributed databases, locking protocol

### 10 Application scaling under shared virtual memory on a cluster of SMPs

Dongming Jiang, Brian O'Kelley, Xiang Yu, Sanjeev Kumar, Angelos Bilas, Jaswinder Pal Singh  
May 1999 **Proceedings of the 13th international conference on Supercomputing**

Full text available:  [pdf\(1.44 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

### 11 Increasing availability under mutual exclusion constraints with dynamic vote reassignment

Daniel Barbara, Hector Garcia-Molina, Annemarie Spauster

November 1989 **ACM Transactions on Computer Systems (TOCS)**, Volume 7 Issue 4

Full text available:  [pdf\(2.53 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Voting is used commonly to enforce mutual exclusion in distributed systems. Each node is assigned a number of votes, and only the group with a majority of votes is allowed to perform a restricted operation. This paper describes techniques for dynamically reassigning votes upon node or link failure, in an attempt to make the system more resilient to future

failures. We focus on autonomous methods for achieving this, that is, methods that allow the nodes to make independent choices about chan ...

12 MGS: a multigrain shared memory system

Donald Yeung, John Kubiawicz, Anant Agarwal

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture**, Volume 24 Issue 2

Full text available:  pdf(1.37 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Parallel workstations, each comprising 10-100 processors, promise cost-effective general-purpose multiprocessing. This paper explores the coupling of such small- to medium-scale shared memory multiprocessors through software over a local area network to synthesize larger shared memory systems. We call these systems Distributed Scalable Shared-memory Multiprocessors (DSSMPs). This paper introduces the design of a shared memory system that uses multiple granularities of sharing, and presents an imp ...

13 Protocols for dynamic vote reassignment

Daniel Barbara, Hector Garcia-Molina, Annemarie Spauster

November 1986 **Proceedings of the fifth annual ACM symposium on Principles of distributed computing**

Full text available:  pdf(783.42 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

14 Mobile computing and applications (MCA): Extending invalid-access prevention policy protocols for mobile-client data caching

Shin Parker, Zhengxin Chen

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available:  pdf(109.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Due to the proliferation of multimedia objects and the subsequent need for managing a large number of multimedia objects within mobile client/server computing environments, there may exist multiple physical copies of the same data object in client caches at the same time with the server as the primary owner of all data objects. This brings new challenges of dealing with caching multimedia data for mobile clients. Invalid-access prevention policy protocols developed in traditional DBMS environmen ...

**Keywords:** invalid-access prevention policy protocol, mobile client, multimedia object, serializability, two phase locking

15 Session 2A: Distributed computing: Deadlock detection in a multidatabase

Randy Appleton

April 1992 **Proceedings of the 30th annual Southeast regional conference**

Full text available:  pdf(559.33 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

Previous results [ACM87] show that the timeout strategy of deadlock detection performs poorly in a single site database. But because of the restrictions imposed on heterogeneous multidatabases, the traditional alternatives are not possible. This paper proposes a new deadlock detection strategy within the restrictions of heterogeneous multidatabases, one that we show performs better than simple timeouts. It is proven that in at least one sense, this algorithm is optimal.

16 Transactional client-server cache consistency: alternatives and performance

Michael J. Franklin, Michael J. Carey, Miron Livny


Full text available:  [pdf\(452.41 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Client-server database systems based on a data shipping model can exploit client memory resources by caching copies of data items across transaction boundaries. Caching reduces the need to obtain data from servers or other sites on the network. In order to ensure that such caching does not result in the violation of transaction semantics, a transactional cache consistency maintenance algorithm is required. Many such algorithms have been proposed in the literature and, as all provide the same ...

17 Doctoral symposium: Encapsulating concurrency with Early-Reply

Scott M. Pike

November 2002 **Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**


Full text available:  [pdf\(232.61 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Component methods often produce their final parameter values long before the method body is ready to terminate. To minimize client blocking, Early-Reply can be used to forward invocation results to the caller as soon as they are (safely) available. After executing Early-Reply, the method remainder and the client caller can proceed concurrently, modulo synchronization constraints. The prime motivation for Early-Reply, then, is to improve performance factors such as response time and resource utilization ...

18 Atomizer: a dynamic atomicity checker for multithreaded programs

Cormac Flanagan, Stephen N Freund

January 2004 **ACM SIGPLAN Notices , Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 39 Issue 1

Full text available:  [pdf\(195.13 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Ensuring the correctness of multithreaded programs is difficult, due to the potential for unexpected interactions between concurrent threads. Much previous work has focused on detecting race conditions, but the absence of race conditions does not by itself prevent undesired thread interactions. We focus on the more fundamental non-interference property of *atomicity*; a method is atomic if its execution is not affected by and does not interfere with concurrently-executing threads. Atomic me ...

**Keywords:** atomicity, dynamic analysis, reduction

19 Quality engineering: How one product team met the quality challenge

Mark Changfoot, Ted Fines, Rod Lawson, Nick Lecic, Pat Morenz, Doug Mortson, Pramod Patel, Shelagh Roberts, Sheik Sahib, Rollie Sing

October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: software engineering - Volume 1**

Full text available:  [pdf\(1.63 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

A primary goal of product development teams today is to ensure that quality is integral to everything they do. This article describes the initiatives the members of one product development team undertook to incorporate quality into every phase of their development process. The product was new, the team was new, and there was a strong commitment from every member of the team to prove just how much one product team could do to ensure the quality of its product. Development methodologies, process ma ...

20 Dynamic voting algorithms for maintaining the consistency of a replicated database

S. Jajodia, David Mutchler

June 1990 **ACM Transactions on Database Systems (TODS)**, Volume 15 Issue 2

Full text available:  pdf(4.07 MB)      Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

There are several replica control algorithms for managing replicated files in the face of network partitioning due to site or communication link failures. Pessimistic algorithms ensure consistency at the price of reduced availability; they permit at most one (distinguished) partition to process updates at any given time. The best known pessimistic algorithm, voting, is a "static" algorithm, meaning that all potential distinguished partitions can be listed in adv ...

Results 1 - 20 of 76

Result page: [1](#) [2](#) [3](#) [4](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)